Segregating event streams and noise with a Markov renewal process model

Dan Stowell dan.stowell@eecs.qmul.ac.uk
Mark D. Plumbley mark.plumbley@eecs.qmul.ac.uk
Centre for Digital Music
Queen Mary University of Londdon

Editor: Editor name

Abstract

We describe an inference task in which a set of timestamped event observations must be clustered into an unknown number of temporal sequences with independent and varying rates of observations. Various existing approaches to multi-object tracking assume a fixed number of sources and/or a fixed observation rate; we develop an approach to inferring structure in timestamped data produced by a mixture of an unknown and varying number of similar Markov renewal processes, plus independent clutter noise. The inference simultaneously distinguishes signal from noise as well as clustering signal observations into separate source streams. We illustrate the technique via a synthetic experiment as well as an experiment to track a mixture of singing birds.

Keywords: Multi-target tracking, clustering, point processes, flow network, sound

1. Introduction

Various approaches exist for the task of inferring the temporal evolution of multiple sources based on joint observations (Mahler, 2007; Van Gael et al., 2008). They are generally based on a model in which sources are continuously observable, in the sense that they are expected to emit/return observations at every time step (though there may be missed detections). Yet there are various types of source for which observations are inherently intermittent, and for which this intermittence exhibits temporal structure that can be characterised as a point process. Examples include sound event sequences such as bird calls or footsteps (Wang and Brown, 2006), internet access logs (Arlitt and Williamson, 1997), pulsars in astronomy (Keane et al., 2010) and neural firing patterns (Bobrowski et al., 2009). Intermittent observations are also often output from *sparse representation* techniques, which transform signals into a representation with activations distributed sparsely in time and state (Plumbley et al., 2010).

In this paper we describe a generic problem setting that may be applied to such data, along with an approach to estimation. We are given a set of timestamped data, and we assume each datum is produced by one of a set of similar but independent signal processes, or by a "clutter" noise process, with known parameters. We do not know the true partitioning of the data into sequences each generated by a single process, and wish to infer this. We do not know how many processes are active, and we do not assume that each process produces the same number of observations, or observations at the same time points.

This specific type of clustering problem has applications in various domains. For example, when sparse representation techniques are used for source separation in time series, they often yield a set of atomic activations which must be clustered according to their underlying source, and preferably to discard any spurious noise activations (Plumbley et al., 2010). Temporal dependence information may help to achieve this (cf. Mysore et al. (2010)). Timestamped data such as internet access logs often contain no explicit user association, yet it may be desirable to group such data by user for for further analysis (Arlitt and Williamson, 1997). In computational audio scene analysis, it is often the case that sound sources emit sound only intermittently during their presence in the scene (e.g. bird calls, footsteps), yet it is desirable to track their temporal evolution (Wang and Brown, 2006).

1.1 Related Work

To our knowledge, this particular problem setting has not been directly addressed in the literature. Temporal data is most commonly treated using a model of sources which update continuously, or synchronously at an underlying temporal sampling rate. Pertinent formulations for our purposes include the infinite factorial hidden Markov model (infinite FHMM) of Van Gael et al. (2008), or the probability hypothesis density filter (PHD filter) (Mahler, 2007), both of which infer an unknown number of independent Markov sources. FHMMs assume that the underlying sources are not intermittent during their lifetime, and also that they persist throughout the whole observation period. Pragmatically, intermittent emissions may be handled by incorporating silence states, though the duration of such states cannot take an arbitrary distribution. The PHD filter allows for stochastic missed detections but not for structured intermittency.

Among techniques which do not assume a synchronous update, graph clustering approaches such as normalised cuts have similarities to our approach (Shi and Malik, 2000). In particular, Lagrange et al. (2008) apply normalized cuts in order to cluster temporally-ordered data. However, the normalised cuts method is applied to undirected graphs, and Lagrange et al. (2008) use perceptually-motivated similarity criteria rather than directed Markov dependencies as considered herein. Further, the normalized cuts method does not include a representation of clutter noise, and so Lagrange et al. (2008) perform signal/noise cluster selection as a separate postprocessing step. In the present work we include an explicit noise model.

Our problem setting also exhibits similarities with that of structure discovery in Bayesian networks (Koivisto and Sood, 2004). However, in that context the dependency structure is inferred from correlations present in multiple observations from each vertex in the structure. In the present case we have only one observation per vertex, plus the partial ordering implied by temporality.

In the following we develop a model in which an unknown number of point-process sources are assumed to be active as well as Poisson clutter, and describe how to perform a maximum likelihood inference which clusters the signal into individual identified tracks plus clutter noise. We then demonstrate the performance of the approach in synthetic experiments, and in an experiment analysing birdsong audio.

2. Preliminaries

Throughout we will consider sets of observations in the form $\{(X, T)\}$ where X is state and T is time. A Markov renewal process (MRP) generates a sequence of such observations having the Markov property:

$$\begin{aligned} &P(\tau_{n+1} \leq t, X_{n+1} = j | (X_1, T_1), \ldots, (X_n = i, T_n)) \\ &\qquad = P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i) \qquad \forall n \geq 1,\, t \geq 0,\, i, j \in \mathcal{S} \end{aligned} \tag{1}$$

where τ_{n+1} is the time difference $T_{n+1} - T_n$. Note that τ is not explicitly given in observations $\{(X, T)\}$, but can be inferred if we know that a particular pair of observations are adjacent members within a sequence.

We will have cause to represent our data as a *network flow* problem (Bang-Jensen and Gutin, 2007, Chapter 3). A *network* is a graph supplemented such that each arc A_{ij} has a *lower capacity* l_{ij} and *upper capacity* u_{ij}, and a *cost* a_{ij}. A *flow* is a function $x : A \to \mathcal{R}_0$ that associates a value with each arc in the network. We will be concerned with integer flows $x : A \to \mathcal{Z}_0$. A flow is *feasible* if $l_{ij} \leq x_{ij} \leq u_{ij}$ for all A_{ij} in the graph, and for all vertices (except for any source/sink vertices) the sum of the inward flow is equal to the sum of the outward flow. For any flow we can calculate a total cost as the sum of $a_{ij}x_{ij}$ over all A_{ij}. We define the *value* of a feasible flow to be the sum of x_{ij} over all arcs leading from source vertices.

The standard terminology of flow networks associates capacities, flows and costs with arcs but not vertices. However, in the following we will have cause to associate such attributes with vertices as well as with arcs. This can be implemented transparently by the standard technique of *vertex expansion*, in which each vertex is replaced by an in-vertex and an out-vertex, plus a single arc between them which bears the associated attributes (Bang-Jensen and Gutin, 2007, Section 3.2.4).

3. Mixtures of Markov Renewal Processes with Clutter Noise

For the present task, we consider MRPs which are time-limited: each process comes into being at a particular point in time (governed by an independent Poisson process with intensity $\lambda_b(X)$), and after each observation it may "die" with an independent death probability $p_d(X)$. Otherwise it transitions to a new random state-and-time according to the transition distribution $f_x(X, \tau)$. The overall system to be considered is not one but a set of such time-limited MRPs, plus a separate Poisson process that generates clutter noise with intensity $\lambda_c(X)$. The MRPs are independent but share common parameters. We will refer to the overall system (including the noise process) as a *multiple Markov renewal process* system or *MMRP*, in order to clarify when we are referring to the whole system or to a single MRP.

We receive a set of N observations in the form $\{(X, T)\}$ and we assume that they were generated by an MMRP for which the process parameters are known, but the number K of MRPs is unknown as well as the allocation of each observation to its generating process. We assume that each observation is generated either by one MRP or by the noise process. Given these observations as well as model parameters $f_x(X, \tau)$, λ_b, p_d, λ_c, there are many ways to cluster the observations into $K \in [0, N]$ non-overlapping subsets to represent the assertion that each cluster represents all the emissions from a single MRP, with H of the

observations not included in any cluster and considered to be noise. The overall likelihood under a chosen clustering is given by

$$\text{likelihood} = \prod_{k=1}^{K} p_{\text{MRP}}(k) \prod_{\eta=1}^{H} p_{\text{NOISE}}(\eta)$$

where $p_{\text{MRP}}(k)$ represents the likelihood of the observation subsequence in cluster k being generated by a single MRP, and $p_{\text{NOISE}}(\eta)$ represents the likelihood of a single observation datum under the noise model. (A set of clusters is arbitrarily indexed by $k \in [1, K]$.)

In order to find the maximum likelihood solution, we may equivalently divide the likelihood expression through by a constant factor, to give an alternative expression to be maximised. We divide by the likelihood that all data were generated by the noise process, to give the likelihood ratio:

$$\text{L} = \prod_{k=1}^{K} \frac{p_{\text{MRP}}(k)}{p_{\text{NOISE}}(k)} \tag{2}$$

where for notational simplicity we use $p_{\text{NOISE}}(k)$ as the joint likelihood of all observations contained within cluster k under the noise model. This likelihood ratio L will shortly be seen to be a convenient expression to optimise.

The component likelihood ratio for a single cluster k is given by

$$\frac{p_{\text{MRP}}(k)}{p_{\text{NOISE}}(k)} = \frac{p_b(X_{k,1}) \cdot p_d(X_{k,n}) \cdot \prod_{i=2}^{n_k} f_{X_{k,i-1}}(X_{k,i}, T_{k,i} - T_{k,i-1})}{\prod_{i=1}^{n_k} p_c(X_{k,i})} \tag{3}$$

where $(X_{k,i}, T_{k,i})$ refers to the ith observation assigned to cluster k, this cluster having n_k observations indexed in ascending time order. $p_d(\cdot)$ refers to the likelihood associated with a single observation under the Poisson process parametrised by λ_d, and similarly for $p_c(\cdot)$ for the clutter process parametrised by λ_c.

The overall likelihood ratio L tells us the relative likelihood that the observation set was generated by the selected clustering of signals and noise, as opposed to the possibility that all observations were generated by clutter noise. Our goal is to find the clustering that yields the highest likelihood ratio, and therefore the set of MRP track identities that is most likely to originate from signal rather than noise.

3.1 Network Flow Representation

For any observation set of non-trivial size, there is a combinatorial explosion of possible clusterings available and enumerating them all is intractable. In this subsection we propose to transform the problem into an equivalent problem of network flow, which can be addressed using graph theoretic techniques.

To maximise the likelihood ratio, we can equivalently minimise its negative logarithm, which we will consider as a "cost" for any particular solution. We define additive component

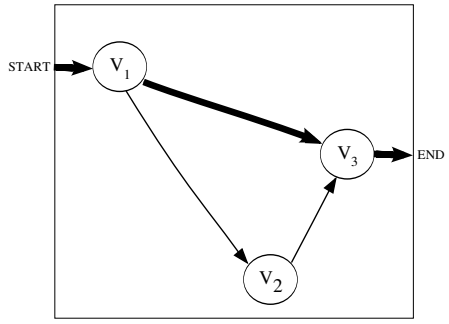

Figure 1: Simple illustration of a path within a network that might correspond to a single MRP sequence. Time increases along the horizontal axis. The bold arrows indicate a path from the first to the third datum (the second datum being left out of the corresponding cluster). The thin arrows indicate an alternative possible path.

costs for birth, death, transition and clutter respectively as:

$$a_b(X) = -\log p_b(X) \tag{4a}$$
$$a_d(X) = -\log p_d(X) \tag{4b}$$
$$a_t(X, X', \tau) = -\log f_X(X', \tau) \tag{4c}$$
$$a_c(X) = \log p_c(X) \tag{4d}$$

which leads to the following expression for the overall cost under a particular cluster assignment:

$$\begin{aligned} -\log(\mathrm{L}) = \sum_{k=1}^{K} \Big(& a_b(X_{k,1}) + a_d(X_{k,n}) \\ & + \sum_{i=2}^{n_k} a_t(X_{ik,i-1}, X_{k,i}, T_{k,i} - T_{k,i-1}) \\ & + \sum_{i=1}^{n_k} a_c(X_{k,i}) \Big) . \end{aligned} \tag{5}$$

The Markov structure of transitions, as well as this representation as additive costs, permit a natural representation as a problem defined on a directed graph. If we construct a directed graph with observations as vertices and possible transitions as arcs, then every possible path in the graph (from any vertex to any other reachable vertex) corresponds to one potential MRP cluster (Figure 1). A set of K paths corresponds to a set of K MRP clusters. To reflect the assumption that each observation is generated by no more than one MRP, we require that a vertex can be a member of no more than one path in such a set. Vertices not included in any of the paths correspond to noise observations.

Given our restriction that a vertex can be included in no more than one path, the problem of finding a mutually compatible set of MRP clusterings is equivalent to solving a

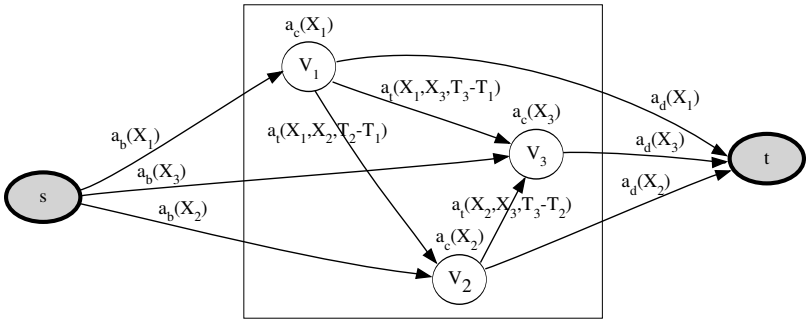

Figure 2: Constructing the weighted flow network for a set of three observations.

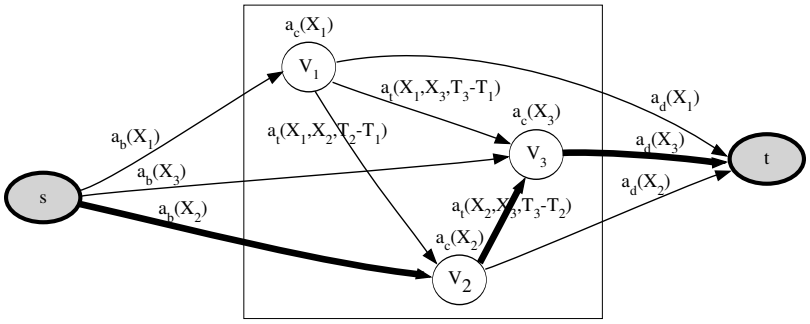

Figure 3: The network of Figure 2, with a single-path flow indicated (*s-2-3-t*).

particular kind of *network flow* problem (Bang-Jensen and Gutin, 2007, Chapter 3). In our case, the concept of a flow will be used to pick out a set of arcs in the graph corresponding to a possible clustering, by associating each arc with a value 1 or 0 indicating whether the arc is included in the clustering. Therefore, in addition to the requirement that the flow is integer-valued, all arcs will be defined to have unit capacity: $l_{ij} = 0, u_{ij} = 1$ for all A_{ij}. To reflect our assumption that each observation can be included in only one cluster, we will also specify unit capacities for all vertices.

It remains to specify how we can associate the costs (4) with the network such that we can solve for the minimum-cost solution to (5). Transition costs will be associated with arcs, and clutter costs with vertices, but in order to include birth and death costs we must modify the network by adding a single "source" vertex with an outward arc to all other vertices, and a single "sink" vertex with an inward arc from all other vertices, and by requiring that no other vertices act as sources or sinks (i.e. in a feasible flow, their inward and outward flows must balance). We then associate birth costs with arcs from the source and death costs with arcs to the sink. This means that all feasible flows in our network will be composed of paths which consist of one single birth cost, plus a sequence of clutter and transition costs, and a single death cost. The source and sink have infinite capacity, allowing for solutions with unbounded K.

Putting these considerations together, constructing the directed graph proceeds as follows:

- A unit-capacity vertex V_i is created corresponding to each observation (X_i, T_i). The clutter noise cost $a_c(X_i)$ is associated with this vertex.

- A unit-capacity arc A_{ij} is created corresponding to each possible transition between two observations such that $T_i < T_j$. The transition cost $a_t(X_i, X_j, T_j - T_i)$ is associated with this arc.

- A "source" vertex s is added, with one arc A_{si} leading from s to each of the observation vertices. The birth cost $a_b(X_i)$ is associated with each arc A_{si}.

- A "sink" vertex t is added, with one arc A_{it} leading from each of the observation vertices to t. The death cost $a_d(X_i)$ is associated with each arc A_{it}.

The temporal ordering of observations means that the graph will contain no cycles.

An illustration of the network constructed for a set of three observations is given in Figure 2. It is clear that any path from the source s to a sink t (we call this an (s,t)-path) visits a sequence of vertices representing a temporal sequence of observations. In the case given in Figure 2, seven different (s,t)-paths are possible, and various combinations of these can form a feasible flow. For example the flow along the single path *s-2-3-t* highlighted in Figure 3 represents the possibility that the observations X_2 and X_3 were generated by a single MRP while X_1 is clutter: the costs associated with flow along that path (the *path flow*) are related to the birth of *2*, the transition from *2* to *3*, and the death of *3*, plus the clutter noise costs. The cost associated with any single-path flow corresponds to one of the K top-level summands in Equation (5). Since in our case each (s,t)-path carries one unit of flow, the *value* of each feasible flow is the number of paths it contains, and corresponds to the number of MRP processes inferred in the data. The total *cost* of each feasible flow is the sum of the path costs contained, and corresponds to the sum calculated in Equation (5).

3.2 Inference

The minimum cost flow in a network constructed according to our scheme corresponds to the clustering with maximum likelihood ratio. So to perform inference we can use existing algorithms that solve minimum-cost network flow problems. The *value* of the minimum-cost flow, which gives the number of MRP sources inferred, may be any integer between 0 and N. We use the Edmonds-Karp algorithm (Bang-Jensen and Gutin, 2007, Chapter 3), which iteratively searches for single paths in a *residual network* representation and does not get trapped in local optima. The Edmonds-Karp algorithm is often used to find maximum-value flow but can be used to optimise cost in our case of binary capacities.

We now consider the time complexity of our inference. The asymptotic time complexity of the Edmonds-Karp search relates to the number of vertices and arcs as $O(|V||A|^2)$. The number of vertices is closely related to the number of observations N; since we generate an arc for every possible transition between a pair of observations, $|A|$ may be on the order of N^2 in the worst case. Hence we add a constraint in constructing the arcs which is reasonable in many applications: we assert that transitions have an upper limit in the size of the time step, and so we do not create arcs for time separations above some threshold $\tau_{\max}$. The

cardinality $|A|$ is then on the order of NB where B is the maximum number of observations within a time window of size $\tau_{\max}$ (and often $B << N$).

If faster search is required at the cost of optimality, greedy search strategies are available. One such strategy is to repeatedly apply a minimum-cost path algorithm to the network, at each iteration taking the resulting path as an identified cluster and removing its vertices from the network before the next iteration. Since the graph is acyclic, finding a minimum-cost path can be performed very efficiently with order $O(|A| + |V|)$ at each iteration (Bang-Jensen and Gutin, 2007, Section 2.3.2); however there is no guarantee of optimality since the overall minimum-cost flow is not guaranteed to be composed of path flows of lowest individual cost. In our experiments we will compare this greedy search empirically against the optimal search.

In the present work we primarily consider offline (batch) inference. However, online inference is possible within the same framework, in which new observations are received incrementally by updating the graph as observations arrive. The Edmonds-Karp search cannot be used on such a dynamic network, except by re-starting the search from scratch upon update. Alternative strategies such as those based on cycle-cancelling can be used to provide an updateable inference (Bang-Jensen and Gutin, 2007, Section 3.10.1). The speed of cycle-cancelling relative to Edmonds-Karp may depend on the nature of the data; we implemented both and found the cycle-cancelling relatively slow.

Thus far we have considered inference using a single set of MMRP model parameters, encoded as the costs in (5). It may be of value to evaluate the same data under different MMRP models, in situations where multiple types of MRP process (having different parameters) may be active. Multiple parametrisations cannot be represented together in a single flow network since they would assign conflicting costs to arcs. To accommodate incompatible costs is equivalent to the "multi-commodity" extension of the minimum-cost flow problem, which is NP-complete (Even et al., 1975). However, if the clutter noise model is held constant between two different MMRP inferences, then the two likelihood ratios calculated by (2) can be divided through to give a likelihood ratio between the two. This allows us to choose between possible MMRP models although not to combine them in a single clustering.

To summarise the MMRP inference described in this section: given a set of observations plus MRP process parameters and noise process parameters, one first represents the data as a flow network, with added source and sink nodes, and with costs representing component likelihoods (Section 3.1). One then applies a minimum-cost flow algorithm to the network such as Edmonds-Karp. Each (s, t)-path in the resulting minimum-cost flow represents a single cluster (a single MRP sequence) in the maximum-likelihood result, while the nodes which receive no flow represent data to be labelled as noise.

4. Experiments

We have described a multiple Markov renewal process (MMRP) inference technique which takes an MRP model, an iid clutter noise model and a set of timestamped data points, and finds a maximum-likelihood partition of the data into zero or more MRP sequences plus clutter noise. In the following, we will illustrate its properties with a synthetic experiment

(Section 4.2), before applying it to a specific task of tracking multiple singing birds in an audio mixture (Section 4.3). We must first consider how to evaluate algorithm outputs.

4.1 Evaluation Measures

To judge the empirical performance of our inference procedure, we must determine whether it can correctly separate signal from noise, and whether it can correctly separate each individual MRP sequence into its own stream. MMRP inference can be considered as a clustering task and could be evaluated accordingly. However, the noise cluster is qualitatively different from the MRP clusters, and the transitions within MRP sequences are the latent features of primary interest, so we will focus our evaluation measures on signal/noise separation and transitions.

In the following our statistics will be based on the standard F-measure (Witten and Frank, 2005, Chapter 5), which summarises precision and recall as follows:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

$$= \frac{2t_+}{(2t_+ + f_- + f_+)} \tag{7}$$

where t_+ is the number of true positive detections, f_+ the number of false positive detections (noise data labelled as signal), and f_- the number of false negative detections (signal data labelled as noise). However, the task for which our MMRP inference is designed is not an ordinary classification task: the signal/noise label for each ground-truth datum can be treated as a class label to be inferred, but the individual signal streams to be recovered do not have labels. To quantify performance we use the F-measure in two ways. The first (which we denote F_{SN}) evaluates the signal/noise classification performance without considering the clustering. The second (which we denote F_{trans}) evaluates the performance at recovering the *pairwise transitions* that are found in the ground-truth signals, i.e. the arcs in the true dependency graph underlying the data.

To illustrate F_{trans}, consider a situation in which a ground-truth sequence was perfectly recovered except that one datum in the middle was left out (Figure 4). This would correspond to a number of true positives, but also two false negatives (the omission of the transition into and out of the missing datum) and one false positive (the mistaken inference of a transition from the missing datum's predecessor to its follower).

Correctly-classified noise observations do not affect F_{trans} since they are not associated with any signal transitions. Thus, F_{SN} is useful to measure signal/noise separation while F_{trans} provides complementary information about correctly recovering separate streams.

4.2 Synthetic Experiment

For our synthetic experiment we generated data in a one-dimensional state space, with dependency structures inspired by the classic "audio streaming" experiments used to explore human auditory grouping of sound sequences (Winkler et al., 2012).

A strictly alternating sequence of the form ABABAB…, where A and B are different tones (Figure 5, top row), can be interpreted either as a single alternating sequence (the "coherent" interpretation) or as a simultaneous but out-of-phase pair of constant sequences

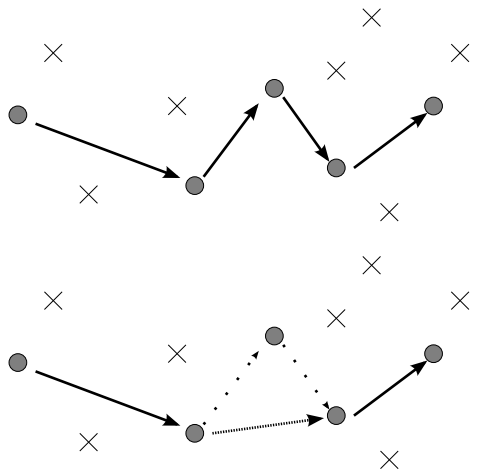

Figure 4: Illustration of errors reflected in F_{trans}. The upper diagram shows a hypothetical ground-truth transition through a sequence of five observations (circles) accompanied by clutter noise (crosses). The lower diagram shows what would happen if inference missed one of those observations out of the chain, resulting in two false-negatives (dotted arrows) for ground-truth transitions not recovered, plus one false-positive (dashed arrow) for a transition that does not exist in the ground-truth. Considering these as well as the two true-positives and applying (7), the F_{trans} value here is $\frac{4}{7}$.

(the "segregated" interpretation). Various factors can lead an observer to prefer one interpretation or the other; here we focus on the case where drift in the timing of the events makes one or the other model more likely (Cusack and Roberts, 2000, Experiment 2). If the sequences drift such that the phase of the As and Bs remain in constant relationship (Figure 5, second row), this is consistent with a "coherent" alternating generator, though may by chance be generated by a "segregated" pair of generators. If the sequences drift such that the phase relationship is not maintained (third row), then this is inconsistent with the "coherent" model but consistent with the "segregated" model. We can generate data with these properties and observe how the MMRP inference behaves under the assumptions of each model.

For our synthetic experiment we defined two separate MRP transition models (one "coherent" and one "segregated") to emit values in a one-dimensional state space $\mathcal{X} \in \mathbb{R}$. Each model was specified by a Gaussian mixture probability distribution defined on state-delta and log-time-delta:

$$\begin{aligned} &P(\tau_{n+1} \leq t, X_{n+1} = j | X_n = i) \\ &\quad = f(X_{n+1} - X_n, \log \tau_{n+1}) \end{aligned} \tag{8}$$

Figure 6 illustrates the transition models. Time differences here are modelled as log-Gaussian to reflect a simple yet perceptually plausible model for lower-bounded time intervals. The variance of the Gaussian components leads to process noise, and the two models

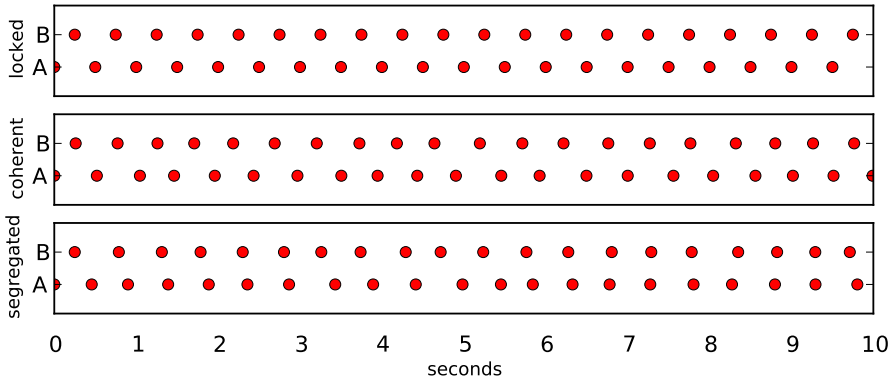

Figure 5: Examples of sequences generated by strict locked ABABAB repetition (top), and by similar generators but with time offsets affected by process noise reflecting either coherent (ABABAB, middle) or segregated (A_A_A_ and _B_B_B, bottom) dependency structure.

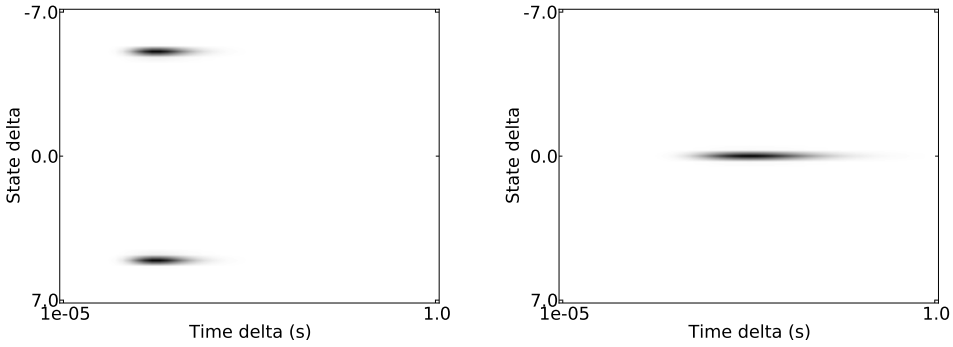

Figure 6: MRP transition probability densities for the two synthetic models: coherent (upper) and segregated (lower).

tend to output different sequences in general. We also define a "locked" model for generation only, which generates a strict ABABAB sequence with no process noise. Its emissions could in principle be explained by either of the two other models.

These models served two roles in our experiment, to synthesise data and to analyse it. For synthesis, we generated one, two or four simultaneous sequences each with a random offset in state space, and we also added iid Poisson clutter noise in the same region of state space, whose intensity is held constant within each run to create a given SNR. In the case of the segregated model, each generator was a pair of such models, independent except for the initial phase and offset, generating As and Bs as was done in Figure 6.

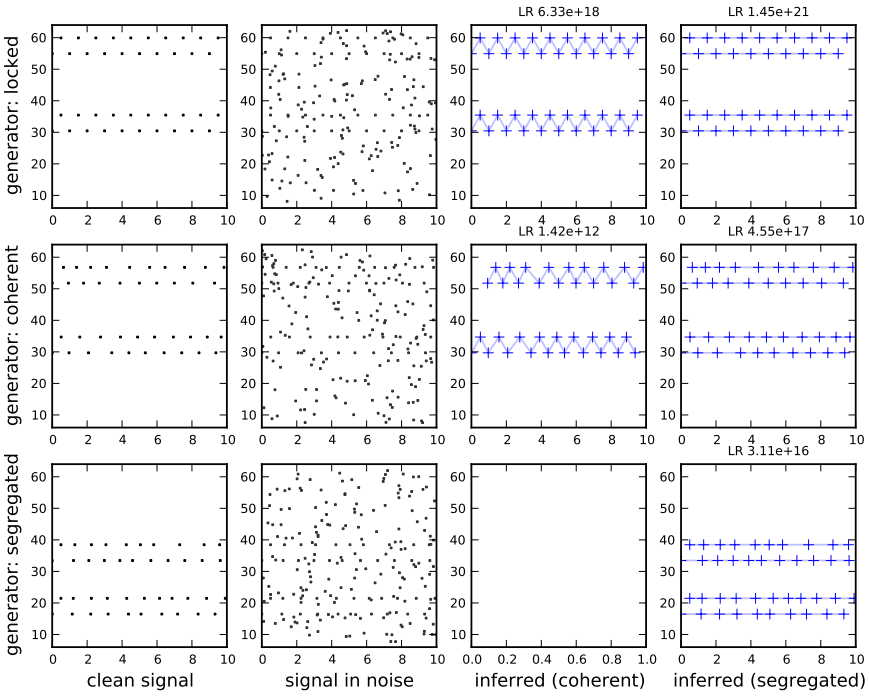

Figure 7: Results of generating observations under the locked, coherent or segregated model (in each row), and then analysing them using the coherent model or the segregated model (final two columns).

The first column of Figure 7 shows the results of generating data under the locked, coherent and segregated models, with two generated sequences present in each case. The second column shows the sequences with added clutter noise at an SNR of -12 dB. The final two columns show the maximum-likelihood signal sequences inferred under the coherent and the segregated model. The MMRP inference typically extracts clear traces corresponding to the ground-truth signals, even in strongly adverse SNR. It is visually evident in the first column that the generated sequences in the middle row have some drift in their rate, but stay in order, while the As and Bs in the bottom row drift relative to each other and do not maintain order. This leads to unlikely emission sequences as judged by the coherent model, and so the coherent model finds the maximum-likelihood solution to be that with no sequences (the blank plot in the figure). Inference using the segregated model extracts traces in all three cases, since the phase-locked drift of the coherent model is not unlikely under the segregated model.

To evaluate our inference procedure, we ran this process multiple times, varying the SNR level, the number of items present, and whether the true SNR was known to the algorithm. When not known, the SNR estimate was arbitrarily held fixed at 0 dB. We tested both the optimal and greedy inference algorithms described in Section 3.2. For each setting we

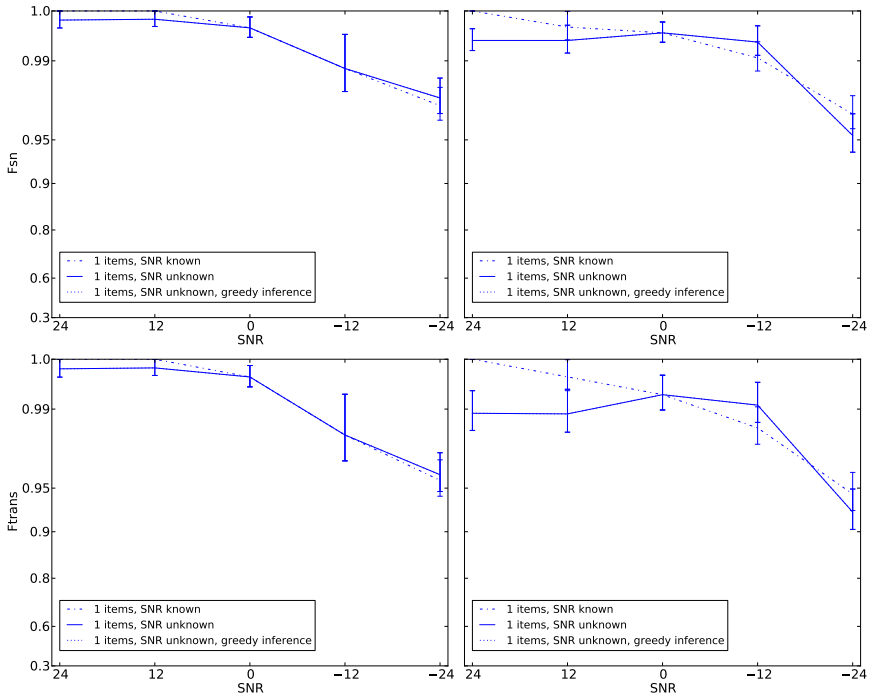

Figure 8: F-measure for signal/noise separation (F_{SN}, upper) and transitions (F_{trans}, lower). The ground truth in each case is a single ABABAB stream, generated via the coherent (left) and segregated (right) cases. Means and standard errors are shown; the vertical axis is reverse-log-scaled so that the results very near 1.0 can be distinguished.

conducted 20 runs and recorded the F_{SN} and F_{trans} statistics. Figures 8 and 9 illustrate the results, and show a consistent pattern according to both statistics. Recovery performance is very strong in all but the most adverse conditions, in most cases being well above 0.95. For these particular scenarios, recovery is impaired under the strongest condition tested (4 simultaneous generators and SNR -24 dB). Under other conditions the recovery is good, whether the true SNR is known to the algorithm or not. Knowing the true SNR does not add a clear improvement to performance, showing that the inference is robust to the SNR estimate parameter. Greedy inference has lower time complexity than the full inference, but when there are multiple streams to be recovered it yields poorer performance than the full algorithm even at very favourable SNR.

4.3 Birdsong Audio Experiment

Many natural sound sources produce signals with structured patterns of emissions and silence, for example birdsong or footsteps. If the emissions due to one such source can be modelled as an MRP, then our inference procedure should be able to separate multiple simultaneous "streams" of emissions. In the following experiment we studied the ability of our inference to perform this separation in data derived from audio signals containing

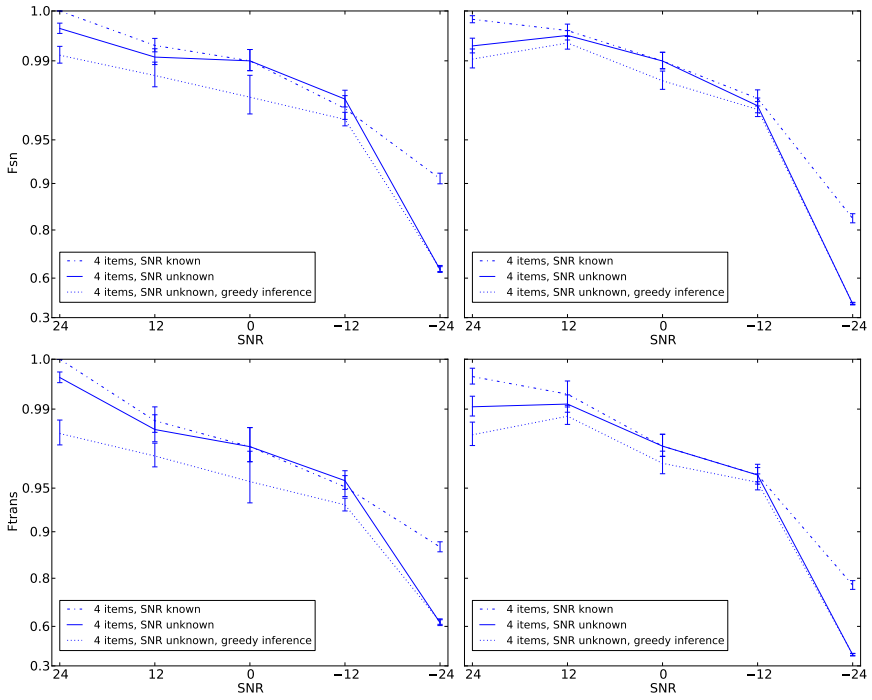

Figure 9: As Figure 8 but with four simultaneous generated streams rather than one.

multiple instances of a species of bird common in many European countries, the Common Chiffchaff (Salomon and Hemim, 1992). Chiffchaff song consists of sequences of typical length 8–20 "syllables". Each syllable is a pitched note consisting of a downward chirp to a briefly-held tone in the region of 5–8 kHz. Syllables are separated by around 0.2–0.3 seconds. The exact note sequence has not to our knowledge been studied in detail; it appears to exhibit only short-range dependency, and is thus amenable to analysis under Markovian assumptions.

4.3.1 Data Preparation

To aid reproducibility, we used recordings from the Xeno Canto database of publicly-available bird recordings.[1] We located 25 recordings of song of the Chiffchaff (species *Phylloscopus collybita*) recorded in Europe (excluding any recordings marked as having "deviant" song or uncertain species identity; also excluding *calls* which are different from *song* in sound and function). The recordings used are listed in Table 1. We converted the recordings to 44.1 kHz mono wave files, high-pass filtered them at 2 kHz, and normalised the amplitude of each file.

Each audio file was analysed separately to create training data; during testing, audio files were digitally mixed in groups of two to five files.

In order to convert an audio file into a sequence of events amenable to MMRP inference, we used spectro-temporal cross-correlation to detect individual syllables of song, as used by Osiejuk (2000). We designed a spectrotemporal template using a Gaussian mixture (GM)

1. `http://www.xeno-canto.org/europe`

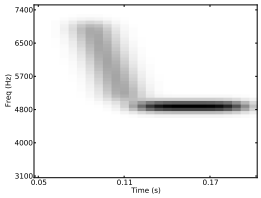

Figure 10: Template used for spectro-temporal cross-correlation detection. The downward and horizontal bars have equal total weight; the latter appears darker because shorter. The template is a manually-constructed Gaussian mixture model having 40 components.

to represent the main characteristics of a single Chiffchaff syllable, a downward chirp to a briefly-held note (Figure 10). The GM was modelled on a Chiffchaff recording from Xeno Canto which was not included in our main dataset (ID number XC48101). Then to analyse an audio file we converted the file into a spectrogram representation (512 samples per frame, 50% overlap between frames, Hann window), and converted the GM to a sampled grid template with the same time-frequency granularity as the spectrogram, before sliding the grid template along the time axis and along the frequency axis (between 3–8 kHz), evaluating the correlation between the template and spectrogram at each location. Correlation values were treated as detections if they were local peaks with value greater than a threshold correlation of 0.8.

Such cross-correlation detection applied to an audio file produces a set of observations, each having a time and frequency offset and a correlation strength (Figure 11). It typically contains one detection for every Chiffchaff syllable, with occasional doubled detections and spurious noise detections. When applied to mixtures of audio, this produces data appropriate for MMRP inference.

In order to derive a Gaussian mixture model (GMM) transition probability model from monophonic Chiffchaff training data, for each audio file in a training set we filtered the observations automatically to keep only the single strongest detection within any 0.2 second window. This time limit corresponds to the lower limit on the rate of song syllables; such filtering is only appropriate for monophonic training sequences and was not applied to the audio mixtures used for testing. The filtered sequences were then used to train a 10-component GMM with full covariance, defined on the vector space having the following four dimensions:

- log(frequency) of syllable one
- log(frequency) of syllable two
- log(magnitude ratio between syllables)
- log(time separation between syllables)

We also trained a separate GMM to create a noise model, taking the set of observations that had been discarded in the above filtering step and training a 10-component GMM with

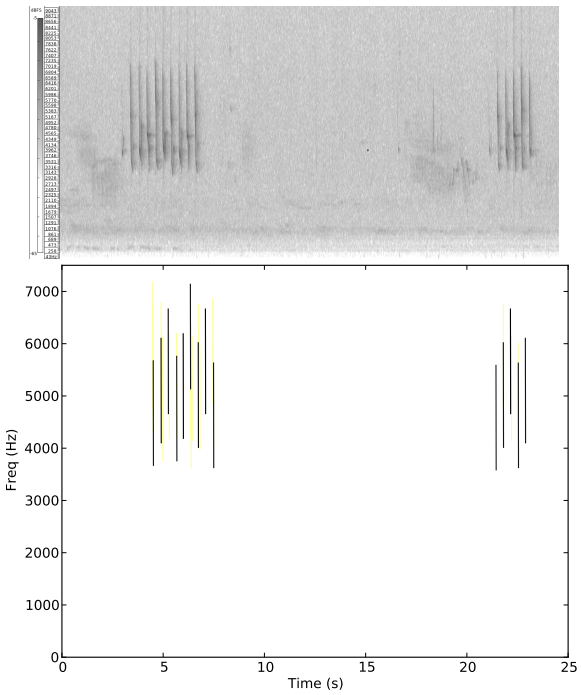

Figure 11: Example of cross-correlation detection: excerpt of spectrogram shown (top), and the corresponding detections (bottom). In the lower image, bold lines represent detections treated as "signal" in the filtering used for training, while the fainter lines represent detections used to train the noise model.

full covariance to fit an iid distribution to the one-dimensional log(frequency) data for the noise observations.

4.3.2 Inference from Audio Mixtures

In order to test whether the MMRP approach could recover syllable sequences from audio mixtures, we performed an experiment using five-fold cross-validation. For each fold we used 20 audio files for training, and then with the remaining five audio files we created audio mixtures of up to five signals, testing recovery in each case.

The quality of signal/noise separation and of clustering the syllables correctly could depend on various features of the experimental task, including whether observations could be extracted from audio mixtures as reliably as from single recordings, the generalisability of the fitted GMMs, noise levels, and the MMRP inference procedure. In order to explore these factors we compared various different analysis approaches:

Audio recovery: The primary approach was to take a mixture audio file, apply spectro-temporal cross-correlation as described above, then to apply MMRP inference using the signal and noise GMMs.

Audio recovery (greedy): This approach was as above, but using greedy recovery rather than the optimal flow inference.

Ideal recovery: There is no guarantee that the same observations will be recovered from the mixture audio as were recovered from the individual recordings. To simulate ideal-case recovery, instead of using the audio mixture we simply pooled the signal and noise observations that had been derived from the test set's individual mono analysis, then performed MMRP inference as in the audio recovery case.

Ideal recovery, synthetic noise: To simulate ideal recovery but with more adverse noise conditions, we proceeded as in the ideal case, but also added extra clutter noise at 0 dB. To do this, we created a copy of every observation in the test set, but assigned it an independent random time position, thus creating noise with the same frequency distribution as the true signal.

Ideal recovery, tested on training set: To measure an "upper limit" on performance and probe the generalisation capability of the algorithm, we proceeded as in the ideal case, but used GMMs trained on the actual test files to be analysed rather than on the separate training data. If this resulted in stronger performance than the ideal-case, it would indicate issues with generalising to signals outside the training set.

Audio recovery, baseline: In order to provide a low-complexity baseline showing the recovery quality using only the marginal properties of the signal and noise, we created a simple baseline system which treated both signal and noise as iid one-dimensional log(frequency) data, using maximum likelihood to label each observation as either signal or noise. The baseline system then clustered together observations that were identified as signal and were separated by less than 0.7 seconds.

We tested each of these approaches using mixtures of one, two, three, four or five of the test recordings. As in the previous experiment, we measured the F_{SN} statistic to evaluate signal/noise separation, and the F_{trans} statistic to evaluate the performance at recovering separate sequences.

Results are shown in Figure 12. Although the two statistics we measure reflect different aspects of performance, they both rank the different analysis approaches in a very similar way. All the MMRP inference runs exhibit a significant and very strong improvement over the baseline. Very strong performance is achieved in the noiseless "ideal recovery" cases, achieving results similar to those in the previous synthetic experiment. The small size of the difference between training on the test data and on the training data indicates that the algorithm can generalise across the data used in our experiment.

When synthetic noise is added to the ideal-recovery case, performance is reduced by a moderate but consistent amount. When we use recovery from audio mixtures, performance reduces again. This shows that the practical task of retrieving detections from audio mixtures has a significant effect on the algorithm performance. However, even in this case our algorithm outperforms the baseline system by a very wide margin, showing the value of MMRP inference for separating signal from noise and clustering signals into MRP streams.

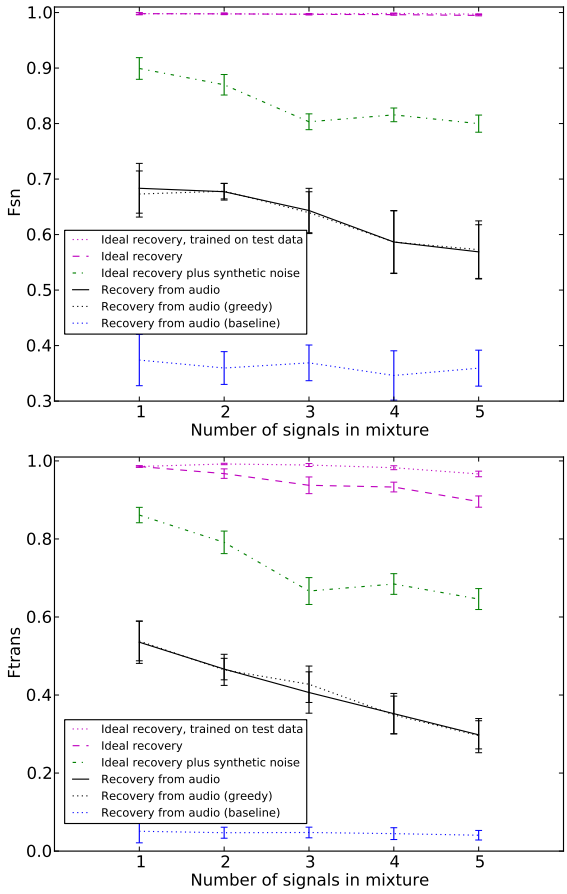

Figure 12: The F_{SN} and F_{trans} evaluation measures for the Chiffchaff audio analyses. Means and standard errors are shown taken over the five folds of the cross-validation.

As we increase the number of recordings in the mixture, performance of all the analysis approaches shows a mild decline. However even with five recordings the performance of the MMRP remains relatively strong.

In this experiment, unlike the previous one, we see very little difference between the performance of the full inference and the greedy inference. Thus the faster greedy inference is appropriate in some but not all situations; in this experiment it is not a limiting factor in performance.

5. Conclusions

In this paper we have introduced a specific clustering problem, that of segregating time-stamped data originating in multiple point processes plus clutter noise. We developed an approach to inferring structure in data produced by a mixture of an unknown number of similar Markov renewal processes (MRPs) plus independent clutter noise. The inference simultaneously distinguishes signal from noise as well as clustering signal observations into

separate source streams, by solving a network flow problem isomorphic to the MMRP mixture problem.

In a synthetic experiment we have shown that inference can perform very well even under high noise conditions (up to -24 dB SNR). In an experiment on birdsong audio data we have also shown strong performance, albeit with a dependence on the quality of the underlying representation to recover events from audio data. Our method is general and has very few free parameters.

The inference in the present work is limited to models without hidden state and with only single-order Markov dependencies. These limitations arise from the combinatorial ambiguity in MMRP mixtures (unlike ordinary Markov models) over which is the immediate predecessor for each observation. Future work will aim to find techniques to broaden the class of models that can be treated in this way.

Acknowledgments

(Acknowledgments to be added in final version.)

References

M. F. Arlitt and C. L. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997. doi: 10.1109/90.649565.

J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Verlag, 1st edition, 2007. URL `http://www.cs.rhul.ac.uk/books/dbook/`.

O. Bobrowski, R. Meir, and Y. C. Eldar. Bayesian filtering in spiking neural networks: Noise, adaptation, and multisensory integration. *Neural Computation*, 21(5):1277–1320, 2009. doi: 10.1162/neco.2008.01-08-692.

R. Cusack and B. Roberts. Effects of differences in timbre on sequential grouping. *Attention, Perception, & Psychophysics*, 62(5):1112–1120, 2000. doi: 10.3758/BF03212092.

S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1975. doi: 10.1109/SFCS.1975.21.

E. F. Keane, D. A. Ludovici, R. P. Eatough, M. Kramer, A. G. Lyne, M. A. McLaughlin, and B. W. Stappers. Further searches for rotating radio transients in the Parkes Multi-beam Pulsar Survey. *Monthly Notices of the Royal Astronomical Society*, 401(2):1057–1068, 2010. doi: 10.1111/j.1365-2966.2009.15693.x.

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004. URL `http://jmlr.csail.mit.edu/papers/v5/koivisto04a.html`.

M. Lagrange, L. G. Martins, J. Murdoch, and G. Tzanetakis. Normalized cuts for predominant melodic source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):278–290, 2008.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston/London, 2007.

G. Mysore, P. Smaragdis, and B. Raj. Non-negative hidden Markov modeling of audio with application to source separation. In *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA / ICA)*, volume 6365/2010, pages 140–148, St. Malo, France, 2010. doi: 10.1007/978-3-642-15995-4_18.

T. S. Osiejuk. Recognition of individuals by song, using cross-correlation of sonograms of Ortolan buntings emberiza hortulana. *Biological Bulletin of Poznań*, 37(1 Suppl):39–50, 2000.

M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2010. doi: 10.1109/JPROC.2009.2030345.

M. Salomon and Y. Hemim. Song variation in the chiffchaffs (phylloscopus collybita) of the western pyreneesthe contact zone between the collybita and brehmii forms. *Ethology*, 92 (4):265–282, 1992. doi: 10.1111/j.1439-0310.1992.tb00965.x.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

J. Van Gael, Y. W. Teh, and Z. Ghahramani. The infinite factorial hidden Markov model. In *Proceedings of Neural Information Processing Systems (NIPS)*, volume 21, 2008.

D. L. Wang and G. J. Brown, editors. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. IEEE Press, 2006.

I. Winkler, S. Denham, R. Mill, T.M. Bőhm, and A. Bendixen. Multistability in auditory stream segregation: a predictive coding view. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1591):1001–1012, 2012. doi: 10.1098/rstb.2011.0359.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition, 2005.

ID	Country	ID	Country
XC103404	pl	XC48263	no
XC25760	dn	XC48383	de
XC26762	se	XC54052	it
XC28027	de	XC55168	fr
XC29706	se	XC56298	de
XC31881	nl	XC56410	ru
XC32011	nl	XC57168	fr
XC32094	no	XC65140	es
XC35097	es	XC77394	dk
XC35974	cz	XC77442	se
XC36603	cz	XC97737	uk
XC36902	nl	XC99469	pl
XC46524	nl		

Table 1: Chiffchaff audio samples used in our dataset, giving the Xeno Canto ID and the country code. Each recording can be accessed via a URL such as `http://www.xeno-canto.org/XC103404`, and the dataset is also archived at `http://archive.org/details/chiffchaff25`